Evaluation of Fuzzy Labor Market by Fuzzy Neural Network

S. Ghavidel, M. Otadi, M. Mosleh

Department of Economics, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran
Department of Mathematics, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran

Abstract: In this paper, a novel hybrid method based on fuzzy neural network for estimate fuzzy coefficients (parameters) of fuzzy supply and demand labor function with fuzzy output and fuzzy inputs, is presented. Here a neural network is considered as a part of a large field called neural computing or soft computing. Moreover, in order to find the approximate parameters, a simple algorithm from the cost function of the fuzzy neural network is proposed. Finally, we illustrate our approach by some numerical examples, specially for Iran labor market.

Key words: Labor market; Fuzzy neural networks; Fuzzy number; Learning algorithm; Fuzzy regression.

INTRODUCTION

General input demand functions generated by considering the firm's profit-maximizing decision can be stated for the two-input case as:

\[ L = L(P, w, r) \]

Where \( L \) is the labor demand, \( P \) is the product price, \( w \) is the labor price (wage) and \( r \) is the capital price. Other way for input demand functions generate is Constant Output Demand Functions with an analysis of cost minimization Cowell (2004) and Henderson (1985), called Shephard's lemma, which uses the q theorem to show that the constant output demand function for \( L \) can be found simply by partially differentiating total costs (C) with respect to \( w \). General labor supply function generated by maximization utility function of consumption and leisure with constant personality budget line, it is written by following:

\[ h = h(p, w, A, e) \]

Where \( h \) is the labor supply, \( p \) is the product price and it has positive effect on \( h \), \( w \) is the wage and its relationship is positive, \( A \) are personality factors such as age, education, gender and etc. \( e \) is the residual term.

In the economy to show the relationship of two or more variables mainly is used regression. For example, in the following equation, fluctuations in \( Y \) are described by \( X_1 \) and \( X_2 \). In most cases \( A_0, A_1 \) and \( A_2 \) are estimated with the OLS technique,

\[ \bar{Y}_i = A_0 + A_1 X_{i1} + A_2 X_{i2} + U_i. \]

Mostly of empirical research for estimation demand and supply of labor are econometrical approach. Heckman and MaCurdy (1980) consider a fixed effects Tobit model to estimate a lifecycle model of female labor supply. Verbon (1980) applies the SUR procedure with one-way error components to a set of four labor demand equations. Conway and Kniesner (1992) who used the Panel Study of Income Dynamics to study the sensitivity of male labor supply function estimates to how the wage is measured. Moffitt (1993) illustrates his estimation method for the linear fixed effects lifecycle model of labor supply using repeated cross-sections from the US Current Population Survey (CPS). Pesaran and Smith (1995) consider the problem of estimating a dynamic panel data model when the parameters are individually heterogeneous and illustrate their results by estimating industry-specific UK labor demand functions. Jorgenson .W (2008) they represent labor demand for each of 35 industrial sectors of the U.S. economy as a response to the prices of productive inputs: labor, capital, and intermediate goods and services. In addition, labor demand is driven by changes in technology.

Numerical solution of a demand and supply of labor is obtained now in a natural way, by extending the existing classical methods to the fuzzy case, we generalized a numerical method presented for approximating demand and supply of labor.

Corresponding Author: M. Otadi, Department of Mathematics, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran
E-mail: otadi@iau.fb.ac.ir; Tel.: +98 9126964202
In this paper, \( Y, X_1, \) and \( X_2 \) have been considered fuzzy variables and the coefficients \( A_0, A_1, \) and \( A_2 \) have been estimated with OLS techniques. In this paper, first, estimated coefficients of regression with three explanatory fuzzy variables and then a numerical example are mentioned for it. Finally, we are estimated supply and demand labor with the assumption of fuzzy variables. The concept of fuzzy numbers and fuzzy arithmetic operations were first introduced by Zadeh (1975), Dubois and Prade (1978). We refer the reader to Kaufmann and Gupta (1985) for more information on fuzzy numbers and fuzzy arithmetic.

Regression analysis is one of the most popular methods of estimation. It is applied to evaluate the functional relationship between the dependent and independent variables. Fuzzy regression analysis is an extension of the classical regression analysis in which some elements of the model are represented by fuzzy numbers. Fuzzy regression methods have been successfully applied to various problems such as forecasting (Chang 1997; Chen and Wang 1999; Tseng and Tzeng 2002) and engineering (Lai and Chang 1994). Thus, it is very important to develop numerical procedures that can appropriately treat fuzzy regression models. Modarres et al. (2005) proposed a mathematical programming model to estimate the parameters of a fuzzy linear regression

\[
Y_i = A_1X_{i1} + A_2X_{i2} + A_nX_{in},
\]

where \( X_{ij} \in \mathbb{R} \) and \( A_1, A_2, \ldots, A_n, Y_i \) are symmetric fuzzy numbers for \( i=1,2,\ldots, m, j=1,2,\ldots, n \). Recently, Mosleh et al. (2010) proposed fuzzy neural network to estimate the parameters of a fuzzy linear regression

\[
Y_i = A_1X_{i1} + A_2X_{i2} + A_nX_{in},
\]

where \( X_{ij} \in \mathbb{R} \) and \( A_1, A_2, \ldots, A_n, Y_i \) are fuzzy numbers for \( i=1,2,\ldots, m, j=1,2,\ldots, n \).


\[
a_1x + a_2x^2 + \ldots + a_nx^n = a_0 \quad \text{where } x \in \mathbb{R} \text{ and } a_0, a_1, \ldots, a_n \text{ are fuzzy numbers, and finding solution to systems of } s \text{ fuzzy polynomial equations (Abbasbandy et al. 2008).}
\]

In this paper, we first propose an architecture of fuzzy neural network (FNN) with fuzzy weights for real input vectors and fuzzy targets to find approximate coefficients to fuzzy linear regression model

\[
\bar{Y}_i = A_0 + A_1X_{i1} + \ldots + A_nX_{in},
\]

where \( i \) indexes the different observations, \( X_{i1}, X_{i2}, \ldots, X_{in} \in \mathbb{R} \), all coefficients and \( \bar{Y}_i \) are fuzzy numbers. The input-output relation of each unit is defined by the extension principle of Zadeh (1975). Output from the fuzzy neural network, which is also a fuzzy number, is numerically calculated by interval arithmetic (Alefeld and Herzberger 1983) for fuzzy weights and real inputs. Next, we define a cost function for the level sets of fuzzy outputs and fuzzy targets. Then, a crisp learning algorithm is derived from the cost function to find the fuzzy coefficients of the fuzzy linear and nonlinear regression models. The proposed algorithm is illustrated by some examples in the last section.

2 Preliminaries:

In this section, the basic notations used in fuzzy calculus are introduced. We start by defining the fuzzy number.

Definition 1:

A fuzzy number is a fuzzy set \( u : \mathbb{R}^1 \to I = [0,1] \) such that

i. \( u \) is upper semi-continuous.

ii. \( u(x) = 0 \) outside some interval \([a, d]\).

iii. There are real numbers \( b \) and \( c \), \( a < b < c < d \), for which

1. \( u(x) \) is monotonically increasing on \([a, b]\),

2. \( u(x) \) is monotonically decreasing on \([c, d]\),

267
3. \( u(x) = 1, b \leq x \leq c \)

The set of all the fuzzy numbers (as given in definition 1) is denoted by \( E^1 \).

An alternative definition which yields the same \( E^1 \) is given by Kaleva (1987), Ming Ma and Friedman (1999).

**Definition 2:**

A fuzzy number \( u \) is a pair \((\underline{u}, \overline{u})\) of functions \( \underline{u}(r) \) and \( \overline{u}(r) \), \( 0 \leq r \leq 1 \), which satisfy the following requirements:

i. \( \overline{u}(r) \) is a bounded monotonically increasing, left continuous function on \((0, 1]\) and right continuous at 0.

ii. \( \underline{u}(r) \) is a bounded monotonically decreasing, left continuous function on \((0, 1]\) and right continuous at 0.

iii. \( \underline{u}(r) \leq \overline{u}(r), 0 \leq r \leq 1. \)

A crisp number \( r \) is simply represented by \( u(\alpha) = \overline{u}(\alpha) = r, 0 \leq \alpha \leq 1. \) The set of all the fuzzy numbers is denoted by \( E^1. \)

A popular fuzzy number is the triangular fuzzy number \( u = (u_m, u_l, u_r) \) where \( u_m \) denotes the modal value and the real values \( u_l > 0 \) and \( u_r > 0 \) represent the left and right fuzziness, respectively. The membership function of a triangular fuzzy number is defined by:

\[
\mu_u(x) = \begin{cases} 
\frac{x - u_m}{u_l} + 1, & u_m - u_l \leq x \leq u_m, \\
\frac{u_m - x}{u_r} + 1, & u_m \leq x \leq u_m + u_r, \\
0, & \text{otherwise.}
\end{cases}
\]

Its parametric form is

\[ u(\alpha) = u_m + u_l (\alpha - 1), \quad \overline{u}(\alpha) = u_m + u_r (1 - \alpha). \]

Triangular fuzzy numbers are fuzzy numbers in \( LR \) representation where the reference functions \( L \) and \( R \) are linear. The set of all triangular fuzzy numbers on \( \mathbb{R} \) is called \( \mathcal{FZ} \).

**2.1 Operations on Fuzzy Numbers:**

We briefly mention fuzzy number operations defined by the extension principle (Zadeh 1975). Since input vector of feedforward neural network is fuzzified in this paper, the operations we use in our fuzzy neural network are fuzzified by means of the extension principle as follows:

\[
\mu_{\Delta+B}(z) = \max\{\mu_A(x) \wedge \mu_B(y) \mid z = x + y\},
\]

\[
\mu_{\Delta B}(z) = \max\{\mu_A(x) \wedge \mu_B(y) \mid z = xy\},
\]

\[
\mu_{f(Ne(x))}(z) = \max\{\mu_{Ne(x)}(x) \mid z = f(x)\},
\]

268
where \(A\), \(B\) and \(Net\) are fuzzy numbers, \(\mu_{\alpha}(\cdot)\) denotes the membership function of each fuzzy number, \(\wedge\) is the minimum operator, and \(f(\cdot)\) is a continuous activation function (such as \(f(x)=x\)) inside units of our fuzzy neural network.

The above operations of fuzzy numbers are numerically performed on level sets. The \(h\)-level set of a fuzzy number \(X\) is defined by

\[
[X]_h = \{ x \in \mathbb{R} \mid \mu_{X}(x) \geq h \} \quad \text{for} \quad 0 < h \leq 1,
\]

and \([X]_0 = \bigcup_{h \in [0,1]} [X]_h\). Since level sets of fuzzy numbers become closed intervals, we denote \([X]_h\) by

\[
[X]_h = [[X]_h^L, [X]_h^U],
\]

where \([X]_h^L\) and \([X]_h^U\) are the lower and the upper limits of the \(h\)-level set \([X]_h\), respectively.

From interval arithmetic (Alefeld 1983), the above operations on fuzzy numbers are written for \(h\)-level sets as follows:

\[
\begin{align*}
A = B & \iff [A]_h = [B]_h \quad \text{for} \quad 0 < h \leq 1, \\
[A + B]_h & = [[A]_h^L + [B]_h^L, [A]_h^U + [B]_h^U], \\
[A.B]_h & = [[A]_h^L \cdot [A]_h^U \cdot [B]_h^L, [B]_h^L \cdot [B]_h^U],
\end{align*}
\]

\(f([Net]_h) = f([[Net]_h^L, [Net]_h^U]) = [f([Net]_h^L), f([Net]_h^U)]\)

where \(f\) is an increasing function. In the case of \(0 \leq [A]_h^L \leq [A]_h^U\), (3) can be simplified as

\[
[A.B]_h = [\min\{[A]_h^L \cdot [B]_h^L, [A]_h^U \cdot [B]_h^U\}, \max\{[A]_h^L \cdot [B]_h^L, [A]_h^U \cdot [B]_h^U\}].
\]

The result of a fuzzy addition of triangular fuzzy numbers is a triangular fuzzy number again. So we only have to compute the following equation:

\[
(a_m, a_l, a_r) + (b_m, b_l, b_r) = (a_m + b_m, a_l + b_l, a_r + b_r)
\]

Considering the fuzzy multiplication, some computational expense problems can be investigated. The result of a fuzzy multiplication is a fuzzy number in \(LR\) representation, but it is difficult to compute the new functions \(L\) and \(R\) because they are not necessarily linear. We approximate this fuzzy multiplication such that it computes a triangular fuzzy number too. This fuzzy multiplication is Denoted by (Feuring 1995). This fuzzy
multiplication is based on the extension principle but is a bit different from the classical fuzzy multiplication. We compute our operation by the following equation:

\[
(a_m, a_l, a_r) \ast (b_m, b_l, b_r) = (c_m, c_l, c_r)
\]

(7)

with

\[
c_m = a_m b_m, c_l = c_m - c_r, c_r = c_\rho - c_m,
\]

\[
c_\lambda := \min(a_\lambda b_\lambda, a_\lambda b_\rho, a_\rho b_\lambda, a_\rho b_\rho),
\]

\[
c_\rho := \max(a_\lambda b_\lambda, a_\lambda b_\rho, a_\rho b_\lambda, a_\rho b_\rho),
\]

where \(a_\lambda = a_m - a_l\) and \(a_\rho = a_m + a_r\). \(a_\lambda\) and \(a_\rho\) denote the left and right limits of the support of fuzzy the number \(a\).

The use of these fuzzy operations has some advantages:

- The distributivity of these operations is retained. This is very important for our theoretical examinations.
- The computational expense is acceptable.
- The idea of fuzzy sets is preserved even if a fuzzy number is characterized by only three values.

We describe the classical definition of distance between fuzzy numbers (Feuring 1995):

**Definition 3:**

The mapping \(\hat{d} : \tilde{FZ} \times \tilde{FZ} \rightarrow \mathbb{R}^+\) is defined by

\[
\hat{d}(A, B) = \max(|a_m - b_m|, |a_\lambda - b_\lambda|, |a_\rho - b_\rho|),
\]

where \(A = (a_m, a_l, a_r)\) and \(B = (b_m, b_l, b_r)\). It can be proved that \(\hat{d}\) is a metric on \(\tilde{FZ}\) and so \((\tilde{FZ}, \hat{d})\) becomes a metric space.

### 2.2 Input-output Relation of Each Unit:

Let us fuzzify a two-layer feedforward neural network with \(n\) input units and one output unit. Input vectors, targets, connection weights and bias are fuzzified (i.e., extended to fuzzy numbers). In order to derive a crisp learning rule, we restrict fuzzy weights, fuzzy inputs and fuzzy target within triangular fuzzy numbers. The input-output relation of each unit of the fuzzified neural network can be written as follows:

**Input units:**

\[
O_{i0} = 1, O_{ij} = X_{ij}, \quad i = 1,2,\ldots,m, \ j = 1,\ldots,n.
\]

(8)

**Output unit:**

\[
Y_i = f(Net_i),
\]

(9)

\[
Net_i = W_0 + O_{i1} \ast W_1 + \ldots + O_{in} \ast W_n, \quad i = 1,2,\ldots,m,
\]

(10)

where \(X_{ij}\) is a fuzzy input and \(W_j\) is the fuzzy weight (see figure 1).

### 2.3 Calculation of Fuzzy Output:

The fuzzy output from each unit in Eqs.(8)-(10) is numerically calculated for level sets of fuzzy inputs and fuzzy weights. The input-output relations of our fuzzy neural network can be written for the \(h\)-level sets:

**Input units:**

(11)

270
Fig. 1: Fuzzy neural network for approximating fuzzy linear regression.

\[ [O_{ij}]_h = [X_{ij}]_h, \quad i = 1, \ldots, m, j = 1, \ldots, n. \]  \hfill (11)

Output unit:

\[ [Y_i]_h = f([Net_i]_h), \]  \hfill (12)

\[ [Net_i]_h = \sum_{j=0}^{n} [O_{ij}]_h [W_{ij}]_h, \quad i = 1, 2, \ldots, m. \]  \hfill (13)

From Eqs.(11)-(13), we can see that the \( h \)-level sets of the fuzzy output \( Y_i \) is calculated from those of the fuzzy inputs and fuzzy weights. From Eqs.(2)-(38), the above relations are written as follows when the \( h \)-level sets of the fuzzy inputs \( a_{ij} \)'s are nonnegative, i.e., \( 0 \leq [X_{ij}]_h^L \leq [X_{ij}]_h^U \) for all \( i, j \)'s:

**Input units:**

\[ [O_{ij}]_h = [[O_{ij}]_h^L, [O_{ij}]_h^U] = [[X_{ij}]_h^L, [X_{ij}]_h^U], \quad i = 1, 2, \ldots, m, j = 1, \ldots, n. \]  \hfill (14)

**Output unit:**

\[ [Y_i]_h = [[Y_i]_h^L, [Y_i]_h^U] = [f([Net_i]_h^L), f([Net_i]_h^U)], \]  \hfill (15)

where \( f \) is an increasing function.

\[ [Net_i]_h^L = \sum_{j=a} [O_{ij}]_h^L [W_{ij}]_h^L + \sum_{j=b} [O_{ij}]_h^L [W_{ij}]_h^U, \]  \hfill (16)

\[ [Net_i]_h^U = \sum_{j=c} [O_{ij}]_h^U [W_{ij}]_h^L + \sum_{j=d} [O_{ij}]_h^U [W_{ij}]_h^U, \quad i = 1, 2, \ldots, m. \]  \hfill (17)
3 The Linear Regression Model:

We have postulated that the dependent fuzzy variable \( Y \) is a function of the independent fuzzy variables \( X_1, X_2, \ldots, X_n \). More formally

\[
f : \mathbb{R}^n \rightarrow E
\]

\[
Y_i = f(X_{i1}, X_{i2}, \ldots, X_{in})
\]

where \( i \) indexes the observations.

The objective is to estimate a fuzzy linear regression (FLR) model, express as follows:

\[
\hat{Y}_i = A_0 + A_1 \hat{X}_{i1} + A_2 \hat{X}_{i2} + \cdots + A_n \hat{X}_{in}.
\]

(18)

When \( f : E^n \rightarrow E \), we might do it by eye-fitting the line that looks best to us. Unfortunately, different people will draw different lines and it would be nice to have a formal method for finding the line that would consistently provide us with the best line possible. What would a "best possible line" look like? Intuitively, it would seem to have to be a line that fit the data well. That is, the distance of the line from the observations should be as small as possible. Let \( A_0, A_1, \ldots, A_n \) denote the list of regression coefficients (parameters). \( A_0 \) is an optional intercept parameter and \( A_1, \ldots, A_n \) are weights or regression coefficients corresponding to \( X_{i1}, \ldots, X_{in} \). Then fuzzy linear regression is given by

\[
\hat{Y}_i = A_0 + A_1 \hat{X}_{i1} + \cdots + A_n \hat{X}_{in},
\]

(19)

where \( i \) indexes the different observations and \( A_0, A_1, \ldots, A_n \) are fuzzy numbers. We are interested in finding \( A_0, A_1, \ldots, A_n \) of fuzzy linear regression such that \( \hat{Y}_i \) approximates \( Y_i \) for all \( i=1,2,\ldots,m \), closely enough according to some norm \( P \), i.e.,

\[
\min P(\hat{Y}_i)_h^\mathcal{L} - [Y_i]_h^\mathcal{L}P \quad \text{and} \quad \min P(\hat{Y}_i)_h^\mathcal{U} - [Y_i]_h^\mathcal{U}P, \quad h \in [0,1].
\]

(20)

Therefore,

\[
\min \hat{d}(\hat{Y}_i, Y_i) \quad \text{for all} \quad i=1,2,\ldots,m.
\]

(21)

Then, it becomes a problem of optimization.

A \( FNN_3 \) (fuzzy neural network with fuzzy input, output signals and fuzzy weights) solution to Eq.(19) is given in figure 1. The input neurons make no change in their inputs and the input signals interact with the weights, so the input to the output neuron is

\[
A_0 + A_1 \hat{X}_{i1} + \cdots + A_n \hat{X}_{in}
\]

and the output, in the output neuron, equals its input, so
How does the FNN solve the fuzzy linear regression? The training data are \{(1, X_{11}, \ldots, X_{1m}), \ldots, (1, X_{m1}, \ldots, X_{mn})\} for inputs and target (desired) outputs are \{Y_1, \ldots, Y_m\}.

We proposed a learning algorithm from the cost function for adjusting fuzzy number weights.

Following section 4, we proposed a learning algorithm such that the network can approximate the fuzzy \(A_0, A_1, \ldots, A_n\) of Eq. (19) to any degree of accuracy.

### 3.1 Learning Fuzzy Neural Network:

Consider the learning algorithm of the two-layer fuzzy feedforward neural network with 2 inputs and one output as shown in figure 1. Let the \(h\)-level sets of the target output \(Y_i, i = 1, \ldots, m\) be denoted

\[
[Y_i]_h = [[Y_{i1}]_h, [Y_{i2}]_h], \quad i = 1, \ldots, n,
\]

where \(Y_{i1}(h)\) shows the left-hand side and \(Y_{i2}(h)\) the right-hand side of the \(h\)-level sets of the desired output.

A cost function to be minimized is defined for each \(h\)-level sets as follows:

\[
[E(W_0, W_1, \ldots, W_n)]_h = [E(W_0, W_1, \ldots, W_n)]^L_h + [E(W_0, W_1, \ldots, W_n)]^U_h,
\]

where

\[
[E(W_0, W_1, \ldots, W_n)]^L_h = \frac{1}{2} \sum_{i=1}^{n} (\overline{Y}_i - [Y_i]^L_h)^2,
\]

\[
[E(W_0, W_1, \ldots, W_n)]^U_h = \frac{1}{2} \sum_{i=1}^{n} (\overline{Y}_i - [Y_i]^U_h)^2.
\]

The total cost function for the input-output pair \((x_i, Y_i)\) is obtained as

\[
e = \sum_h [E(W_0, W_1, \ldots, W_n)]_h.
\]

Hence \([E(W_0, W_1, \ldots, W_n)]^L_h\) denotes the error between the left-hand sides of the \(h\)-level sets of the desired and the computed output, and \([E(W_0, W_1, \ldots, W_n)]^U_h\) denotes the error between the right-hand sides of the \(h\)-level sets of the desired and the computed output.

In the research of neural networks, the norm \(P, P\) is often defined as follows:
\begin{equation}
[E(W_0, W_1, \ldots, W_n)]_h^L = \frac{1}{2} \sum_{i=1}^{m} (\bar{Y}_i)_h^L - [Y_i]_h^L)^2 = \frac{1}{2} \sum_{j=0}^{n} \left( \sum_{j=0}^{n} [O_{ij}^j * W_j]_h^L - [Y_i]_h^L \right)^2,
\end{equation}

\begin{equation}
[E(W_0, W_1, \ldots, W_n)]_h^U = \frac{1}{2} \sum_{i=1}^{m} (\bar{Y}_i)_h^U - [Y_i]_h^U)^2 = \frac{1}{2} \sum_{j=0}^{n} \left( \sum_{j=0}^{n} [O_{ij}^j * W_j]_h^U - [Y_i]_h^U \right)^2.
\end{equation}

Clearly, this is a problem of optimization of quadratic functions without constraints that can usually be solved by gradient descent algorithm. In fact, denoting

\[ \nabla E(W) |^L_h = \left( \frac{\partial E(W)}{\partial W_0}, \ldots, \frac{\partial E(W)}{\partial W_n} \right)^T, \]

\[ \nabla E(W) |^U_h = \left( \frac{\partial E(W)}{\partial W_0}, \ldots, \frac{\partial E(W)}{\partial W_n} \right)^T, \]

in order to solve equation (36), assume \( k \) iterations to have been done and get the \( k \)th iteration point \( W_k \).

REMARK 1. Since the equations (25) are quadratic functions, supposing \( 0 \leq [O_{ij}^j]_h^L \leq [O_{ij}^j]_h^U \) and \( 0 \leq [W_j]_h^L \leq [W_j]_h^U \), for \( i, \ldots, m, j = 0, \ldots, n \), we rewrite these as follows:

\[ [E(W)]_h^L = \frac{1}{2} \sum_{j=0}^{n} \sum_{j=0}^{n} [X_{ij}^j * W_j]_h^L - [Y_i]_h^L)^2 \]

\[ = \frac{1}{2} \sum_{j=0}^{n} \sum_{j=0}^{n} \left( \sum_{j=0}^{n} [X_{ij}^j]_h^L [W_j]_h^L \right)^2 - 2 [Y_i]_h^L \sum_{j=0}^{n} [X_{ij}^j]_h^L [W_j]_h^L + ([Y_i]_h^L)^2 \]

\[ = \frac{1}{2} \left\{ \sum_{j=0}^{n} ([X_{ij}]_h^L)^2 ([W_j]_h^L)^2 + \sum_{j=0}^{n} ([X_{ij}]_h^L)^2 ([W_j]_h^L)^2 + \ldots \right\} \]

\[ + \sum_{j=0}^{n} ([X_{ij}]_h^L)^2 ([W_j]_h^L)^2 + 2 \sum_{j=0}^{n} [X_{ij}]_h^L [X_{i2}]_h^L [W_j]_h^L + \ldots + 2 \sum_{j=0}^{n} [X_{ij}]_h^L [X_{in}]_h^L [W_j]_h^L \]

\[ + 2 \sum_{j=0}^{n} [X_{i1}]_h^L [X_{i2}]_h^L [W_j]_h^L + \ldots + 2 \sum_{j=0}^{n} [X_{i1}]_h^L [X_{in}]_h^L [W_j]_h^L [W_n]_h^L \]
\[
+2(\sum_{j=1}^{m}[X_{12}]^k_h[X_{13}]^l_h[W_2]^k_j h + 2(\sum_{j=1}^{m}[X_{12}]^k_h[X_{14}]^l_h)[W_2]^k_j h
\]

\[+\ldots + 2(\sum_{j=1}^{m}[X_{12}]^k_h[X_{1n}]^l_h)[W_2]^k_j h + \ldots +
\]

\[
2(\sum_{j=1}^{m}[X_{12}]^k_h[X_{1n}]^l_h)[W_2]^k_j h + (\sum_{j=1}^{m}[X_{12}]^k_h[Y_{1j}]^l_h)[W_2]^k_j h
\]

\[+(-\sum_{j=1}^{m}[X_{12}]^k_h[Y_{1j}]^l_h)[W_2]^k_j h + \ldots + (-\sum_{j=1}^{m}[X_{1n}]^k_h[Y_{1j}]^l_h)[W_2]^k_j h + \frac{1}{2}\sum_{j=1}^{m}(Y_{1j})^2
\]

\[= \frac{1}{2} \left([W_2]^l_h\right)^T[Q]^l_h[W_2]^l_h + ([B]^l_h)^T[W_2]^l_h + [C]^l_h,
\]

where

\[[Q]^l_h = ([q_j])^l_h,
\]

\[L_h = ([b_1]^l_h,\ldots,[b_n]^l_h)^T,
\]

\[L_h = \frac{1}{2}\sum_{j=1}^{m}(Y_{1j})^2,
\]

\[L_h = \sum_{k=1}^{m}[X_{1k}]^l_h[X_{1k}]^l_h,
\]

with \([q_j]^l_h = [q_j]^l_h\) and \([b_i]^l_h = -\sum_{k=1}^{m}[X_{1k}]^l_h[Y_{1j}]^l_h\). We have

\[
[V_{\nabla}(W)]^l_h = [Q]^l_h[W]^l_h + [B]^l_h,
\]

and

275
\[
[E(W)]_h^U = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=0}^{n} [X_{ij}]^U_h^*[W_{ij}]_h^U - [Y_i]_{h_i}^U)^2
\]

\[
= \frac{1}{2} \sum_{i=1}^{m} ((\sum_{j=0}^{n} [X_{ij}]^U_h^*[W_{ij}]_h^U)^2 - 2[Y_i]_{h_i}^U \sum_{j=0}^{n} [X_{ij}]^U_h^*[W_{ij}]_h^U + ([Y_i]_{h_i}^U)^2)
\]

\[
= \frac{1}{2} \left\{ \left( \sum_{i=1}^{m} ([X_{i1}]_h^U)^2 \right) ([W_{11}]_h^U)^2 + \left( \sum_{i=1}^{m} ([X_{i2}]_h^U)^2 \right) ([W_{21}]_h^U)^2 + \ldots \right\}
\]

\[
+ \left( \sum_{i=1}^{m} ([X_{in}]_h^U)^2 \right) ([W_{1n}]_h^U)^2 + 2 \left( \sum_{i=1}^{m} [X_{i1}]_h^U [X_{i2}]_h^U [W_{11}]_h^U [W_{21}]_h^U + \ldots + 2 \left( \sum_{i=1}^{m} [X_{i1}]_h^U [X_{in}]_h^U [W_{11}]_h^U [W_{in}]_h^U
\]

\[
+ \ldots + 2 \left( \sum_{i=1}^{m} [X_{i2}]_h^U [X_{in}]_h^U [W_{12}]_h^U [W_{in}]_h^U
\]

\[
2 \left( \sum_{i=1}^{m} [X_{i1}]_h^U [X_{in}]_h^U [W_{n1}]_h^U + (- \sum_{i=1}^{m} [X_{i1}]_h^U [Y_i]_h^U)([W_1]_h^U
\]

\[
+ (- \sum_{i=1}^{m} [X_{i2}]_h^U [Y_i]_h^U)([W_2]_h^U + \ldots + (- \sum_{i=1}^{m} [X_{in}]_h^U [Y_i]_h^U)([W_n]_h^U + \frac{1}{2} \sum_{i=1}^{m} ([Y_i]_h^U)^2
\]

\[
= \frac{1}{2} ([W]_h^U)^T [Q]_h^U [W]_h^U + ([B]_h^U)^T [W]_h^U + [C]_h^U,
\]

where
\[ [Q]^U_h = [(q_{ij})]^U_h, \]

\[ U^U_h = ([b_{0j}]^U_h, \ldots, [b_{nj}]^U_h)^T, \]

\[ U^U_h = \frac{1}{2} \sum_{i=1}^{m} ([Y]^U_i)_h^2, \]

\[ U^U_h = \sum_{k=1}^{m} [X^U_{ki}]_h [X^U_{kj}]_h, \]

with \[ [q_{ij}]_h^U = [q_{ij}]_h \] and \[ [b_{ij}]_h^U = -\sum_{k=1}^{m} [X^U_{ki}]_h [Y]^U_i _h \]. We have

\[ [\nabla E(W)]^U_h = [Q]^U_h [W]^U_h + [B]^U_h, \quad \text{(27)} \]

To find the stationary point of \([E(W)]_h^U = ([E(W)]_h^U, [E(W)]_h^U)\), we should put

\[ [\nabla E(W)]_h^U = [\nabla E(W)]_h^U = 0 \iff (0, 0, \ldots, 0)^T. \]

When \([Q]^U_h\) and \([Q]^U_h\) are positive definite matrices, the stationary point can be obtained as follows:

\[ [W^*_h]^U = -([Q]^U_h)^{-1} [B]^U_h, \quad \text{(28)} \]

\[ [W^*_h]^U = -([Q]^U_h)^{-1} [B]^U_h. \]

The Hessian matrices at this point are

\[ [\nabla^2 E(W^*_h)]_h^U = [\nabla (\nabla E(W^*_h))]_h^U = \]

\[
\begin{bmatrix}
\left[ \frac{\partial^2 E(W)}{\partial W^2_i} \right]_h^L & \left[ \frac{\partial^2 E(W)}{\partial W^2_i \partial W^2_j} \right]_h^L & \cdots & \left[ \frac{\partial^2 E(W)}{\partial W^2_i \partial W^2_n} \right]_h^L \\
\vdots & \vdots & \ddots & \vdots \\
\left[ \frac{\partial^2 E(W)}{\partial W^2_j \partial W^2_n} \right]_h^L \\
\left[ \frac{\partial^2 E(W)}{\partial W^2_n \partial W^2_n} \right]_h^L \\
\end{bmatrix}
= [Q]^U_h,
\]

and

277
\[ [\nabla^2 E(W')]_h^\nu = [\nabla (\nabla E(W'))]_h^\nu = \\
\begin{bmatrix}
\frac{\partial^2 E(W)}{\partial W_1^2} & \frac{\partial^2 E(W)}{\partial W_1 \partial W_2} & \ldots & \frac{\partial^2 E(W)}{\partial W_1 \partial W_n} \\
\frac{\partial^2 E(W)}{\partial W_2 \partial W_1} & \frac{\partial^2 E(W)}{\partial W_2^2} & \ldots & \frac{\partial^2 E(W)}{\partial W_2 \partial W_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 E(W)}{\partial W_n \partial W_1} & \frac{\partial^2 E(W)}{\partial W_n \partial W_2} & \ldots & \frac{\partial^2 E(W)}{\partial W_n^2}
\end{bmatrix}_{n \times n}
\]

which are positive definite matrices because \([Q]_h^L\) and \([Q]_h^U\) are positive definite. From optimization theory, we known that \([W^*]_h = ([W^*]_h^L, [W^*]_h^U) = (-[Q]^{-1}_h^L [B]_h^L, -[Q]^{-1}_h^U [B]_h^U)\), is the unique solution of the problem.

REMARK 2. The above method is not very convenient in applications. Now we consider its explicit scheme.

Since \([\nabla^2 E(W)]_h^L = [Q]_h^L [W]_h^L + [B]_h^L\) and \([\nabla^2 E(W)]_h^U = [Q]_h^U [W]_h^U + [B]_h^U\), then

\([\nabla^2 E(W_k)]_h^L = [Q]_h^L [W_k]_h^L + [B]_h^L\) and \([\nabla^2 E(W_k)]_h^U = [Q]_h^U [W_k]_h^U + [B]_h^U\). We know that (Li et al. 2003).

\([([\nabla^2 E(W)]_h^L)]_h^T [\nabla^2 E(W_k)]_h^L = 0, \quad ([\nabla^2 E(W_k)]_h^U)]_h^T [\nabla^2 E(W_k)]_h^L = 0,\)

to therefore we have (Ishibuchi et al. 1995).

\([([Q]_h^L [W_k]_h^L - [\mu_k]_h^L [\nabla^2 E(W_k)]_h^L]_h^T + [B]_h^L)]_h^T ([Q]_h^L [W_k]_h^L + [B]_h^L) = 0\)

and

\([([Q]_h^U [W_k]_h^U - [\mu_k]_h^U [\nabla^2 E(W_k)]_h^U]_h^T + [B]_h^U)]_h^T ([Q]_h^U [W_k]_h^U + [B]_h^U) = 0.\)

Rearranging them, we have:

\([([\nabla^2 E(W_k)]_h^L)]_h^T - [\mu_k]_h^L [Q]_h^L [\nabla^2 E(W_k)]_h^L)]_h^T [\nabla^2 E(W_k)]_h^L = 0,\)

and

\([([\nabla^2 E(W_k)]_h^U)]_h^T - [\mu_k]_h^U [Q]_h^U [\nabla^2 E(W_k)]_h^U)]_h^T [\nabla^2 E(W_k)]_h^U = 0.\)

From these equations, we can easily get an expression for \([\mu_k]_h^L\) and \([\mu_k]_h^U:\)

\[\begin{align*}
[\mu_k]_h^L &= \frac{([\nabla^2 E(W_k)]_h^L)]_h^T [\nabla^2 E(W_k)]_h^L)}{([\nabla^2 E(W_k)]_h^L)]_h^T [Q]_h^L [\nabla^2 E(W_k)]_h^L) \\
[\mu_k]_h^U &= \frac{([\nabla^2 E(W_k)]_h^U)]_h^T [\nabla^2 E(W_k)]_h^U)}{([\nabla^2 E(W_k)]_h^U)]_h^T [Q]_h^U [\nabla^2 E(W_k)]_h^U) .
\end{align*}\]
Substituting these into equations (Ishibuchi et al. 1995; Ishibuchi and Nii 2001; Rumelhart et al. 1986), we obtain

\[ W_{k+1} = W_k + \Delta W_k, \]
\[ \Delta W_k = -\mu_k \nabla E(W_k) + \alpha \Delta W_{k-1}, \]  

where \( k \) indexes the number of adjustments, \( \mu_k = ([\mu_k]^U, [\mu_k]^L) \) is a learning rate and \( \alpha \) is a constant momentum term (a positive real number).

We can also obtain similar relations for \( [X_{y_i}]^U \leq [X_{y_i}]^L \leq 0 \) and \( [W_j]^U \leq [W_j]^L \leq 0 \) \( i=1,2,\ldots,m \), \( j=0,\ldots,n \), and other cases.

4 The Nonlinear Regression Model:

We have postulated that the dependent fuzzy variable \( Y \), is a function of the independent fuzzy variables \( X_1, X_2,\ldots, X_n \). More formally

\[ f : E^n \rightarrow E \]
\[ Y_i = f(X_{i1}, X_{i2},\ldots, X_{in}) \]

where \( i \) indexes the observations.

The objective is to estimate a fuzzy nonlinear regression (FNLR) model, express as follows:

\[ \bar{Y}_i = A_0 \hat{\ast} X_{i1}^A \hat{\ast} X_{i2}^A \hat{\ast} \cdots \hat{\ast} X_{in}^A. \]  

where \( i \) indexes the different observations, \( A_0, A_1,\ldots, A_n \) are fuzzy numbers and \( X_{i1},\ldots, X_{in} \) are fuzzy numbers. Let \( A_0, A_1,\ldots, A_n \) denote the list of regression coefficients (parameters), \( A_0 \) is an optional intercept parameter and \( A_1,\ldots, A_n \) are weights or regression coefficients corresponding to \( X_{i1},\ldots, X_{in} \).

By the above equation, we have

\[ Ln\bar{Y}_i = Ln(A_0) + A_1 \hat{\ast} LnX_{i1} + \ldots + A_n \hat{\ast} LnX_{in}, \]  

where \( Ln \) is an increasing function. From interval arithmetic [?], the above operations on fuzzy numbers are written for \( h \)-level sets as follows.

Input units:

\( O_{i0} = 1, O_{ij} = X_{ij}, \quad j = 0,1,\ldots,n, i = 1,\ldots,m. \)

Output unit:

\[ Ln[\bar{Y}_i]^L = [Ln[\bar{Y}_i]^L, Ln[\bar{Y}_i]^U] = [f([Net_i]^L), f([Net_i]^U)], \]

where \( f(x) = x \) is an increasing function.
\[
[\text{Net}_i^L]_h = \sum_{j \in a} [O_{ij}]_h^L [W_{ji}]_h^L + \sum_{j \in b} [O_{ij}]_h^L [W_{ji}]_h^U, \quad (34)
\]
\[
[\text{Net}_i^U]_h = \sum_{j \in c} [O_{ij}]_h^U [W_{ji}]_h^L + \sum_{j \in d} [O_{ij}]_h^U [W_{ji}]_h^L, \quad i = 1, 2, \ldots, m, \quad (35)
\]

for \([W_{ji}]_h^U \geq [W_{ji}]_h^L \geq 0\), where \([A_0]_h^L, [A_0]_h^U = \begin{bmatrix} e^{[W_{0i}]_h^L}, e^{[W_{0i}]_h^U} \end{bmatrix}, \quad a = \{ j \mid [O_{ij}]_h^L \geq 0 \}, \quad b = \{ j \mid [O_{ij}]_h^U < 0 \}, \quad c = \{ j \mid [O_{ij}]_h^U \geq 0 \}, \quad d = \{ j \mid [O_{ij}]_h^U < 0 \}, \quad a \cup b = \{ 0, \ldots, n \} \quad \text{and} \quad c \cup d = \{ 0, \ldots, n \}.

We are interested in finding \(A_0, A_1, \ldots, A_n\) of fuzzy nonlinear regression such that \(\overline{Y}_i\) approximates \(Y_i\) for all \(i = 1, 2, \ldots, m\), closely enough according to some norm \(P, P\), i.e.,
\[
\min P[\text{Ln}\overline{Y}_i]_h^L - [\text{Ln}Y_i]_h^L P \quad \text{and} \quad \min P[\text{Ln}\overline{Y}_i]_h^U - [\text{Ln}Y_i]_h^U P, \quad h \in [0, 1].
\]

Therefore,
\[
\min \hat{d}(\text{Ln}\overline{Y}_i, \text{Ln}Y_i) \quad \text{for all} \quad i = 1, 2, \ldots, n. \quad (36)
\]

Then, it becomes a problem of optimization. A FNN for solving Eq. (32) is given in figure 2.

5 Comparison with Other Methods:

This study would not be completed without comparing it with other existing methods. Some comparisons are as follows:

- Mosleh et al. (2010) have considered the fuzzy regression \(Y_i = A_0 + A_1 x_{i1} + A_2 x_{i2} + A_n x_{in}\) where \(x\) is a crisp and \(y\) is a fuzzy number, but in this paper, the input and output are both fuzzy numbers. For more details see example 6.2.

- In this paper, if we take the fuzzy inputs as fuzzy points, we will have \([x]_h = [[x]^L]_h^L, [x]^U]_h^U\) and in this case, the content of the present paper reduces to that of (Mosleh et al. 2010). From the point of view of prediction, we have done this comparison between this paper and (Kao et al. 2003; Tanaka et al. 1989) in example 6.1.

6 Numerical Examples:

In this section we have two example, one of them for show useful this technique and another is numerical example from Iran labor market data:

Example 6.1:

Kao et al. (2003) and Tanaka et al. (1989) used an example to illustrate their regression model, in that the explanatory variable is crisp and the responses are triangular fuzzy numbers. That example has five sets of the \((x_i, Y_i)\) observations, see table 1. For each fuzzy numbers, we use 11 h-cuts \(h = 0, 0.1, \ldots, 1\), where we calculate the error of each fuzzy output by
\[
e_{Y_i} = \frac{1}{2} \sum_h ([\hat{Y}_i]_h^L - [Y_i]_h^L)^2 + \frac{1}{2} \sum_h ([\hat{Y}_i]_h^U - [Y_i]_h^U)^2,
\]
and total error by Eq. (24).
In the computer simulation of this example, we use the following specifications of the learning algorithm.

(1) Number of input units: 2 units.

(2) Number of output units: 1 unit.

(3) Stopping condition: $K = 8$ iterations of the learning algorithm.

The training starts with $W_1(1) = (1,0.5,0.5)$ and $W_1'(1) = (0.3,0.3,0.2)$. Applying the proposed method to the approximate solution of problem (19). In symbols, the fuzzy neural network model is:

$$Y_{FNN} = (4.9499,1.8399,1.8398) + (1.71,0.16,0.1601)x_i.$$  

In the study of Tanaka et al. (1989), the results of the Min problem of $h=0$ is used for comparison. The fuzzy regression model is:

$$Y_T = (3.850,3.850,3.850) + 2.100x_i.$$  

In the study of Kao et al. (2003), the results of the fuzzy regression model is:

$$Y_K = 4.808 + 1.718x_i + (0.118,2.32,2.32).$$

To compare the performance of these three methods in estimation, we apply to calculate the errors in estimating the observed responses. Table 1 shows the errors in estimating the five observation for these three methods. The total error of the fuzzy neural network method is 79.0118, which is obviously better than the total error of 81.6567 calculated from the Kao method and the total error of 186.8226 calculated from the Tanaka method. Figure 3 depicts the estimations of these three methods.

![Fuzzy neural network for approximating fuzzy nonlinear regression.](image)

**Table 1:** Numerical data and the estimation errors for example 6.1.

<table>
<thead>
<tr>
<th>Independent variable</th>
<th>Response variable</th>
<th>Errors in estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Tanaka</td>
</tr>
<tr>
<td>1</td>
<td>$(8.0,1.8,1.8)$</td>
<td>62.4071</td>
</tr>
<tr>
<td>2</td>
<td>$(6.4,2.2,2.2)$</td>
<td>62.4071</td>
</tr>
<tr>
<td>3</td>
<td>$(9.5,2.6,2.6)$</td>
<td>10.6631</td>
</tr>
<tr>
<td>4</td>
<td>$(13.5,2.6,2.6)$</td>
<td>23.2031</td>
</tr>
<tr>
<td>5</td>
<td>$(13.0,2.4,2.4)$</td>
<td>28.1421</td>
</tr>
<tr>
<td>Total error</td>
<td></td>
<td>186.8226</td>
</tr>
</tbody>
</table>
Example 6.2:
Labor Demand and Supply of Iran:

In this example we used supply and demand data of Iran labor market. Table 2 and 3 are shown demand and supply data.

Consider the fuzzy data for a dependent fuzzy variable $Y$ that it is total employment in Iran and two independent fuzzy variables $x_1$ and $x_2$ that indicated GDP and wage respectively in table 2. Also in table 3, $Y$ is labor force participation rate and $x_1$ and $x_2$ indicated same in table 2.

Using these data, develop an estimated fuzzy regression equation

$$\hat{Y}_i = \ln(A_0) + A_1 \hat{x}_1 \ln x_{1i} + A_2 \hat{x}_2 \ln x_{2i} + A_3 \hat{x}_3 \ln Y_{i-1,n}.$$  

In the computer simulation of this example, we use the following specifications of the learning algorithm.

1. Number of input units: 3 units.
2. Number of output units: 1 unit.

The training starts with $W_0(1) = (1, 0.25, 0.25), W_1(1) = (0.5, 0.25, 0.25), W_2(1) = (-0.5, 0.25, 0.25)$ and $W_3(1) = (0.5, 0.25, 0.25)$. Applying the proposed method to the approximate solution of problem (33). In symbols, the fuzzy neural network model is:
Table 2: Inputs and output data for labor demand estimate in example 6.2.

<table>
<thead>
<tr>
<th>Year</th>
<th>$X_{i1}$</th>
<th>$X_{i2}$</th>
<th>$Y_{i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1988</td>
<td>(180822.5,0.75,1.25)</td>
<td>(19050.6,1,1)</td>
<td>(11618745.7,0.5,0.25)</td>
</tr>
<tr>
<td>1989</td>
<td>(191502.6,1.2,1.1)</td>
<td>(19264.1,0.75)</td>
<td>(12547725.9,0.25,2)</td>
</tr>
<tr>
<td>1990</td>
<td>(215387.1,1.14)</td>
<td>(20611.4,0.75)</td>
<td>(13096558.2,0.75)</td>
</tr>
<tr>
<td>1991</td>
<td>(245036.4,1.2,0.75)</td>
<td>(22778.4,1,0.25)</td>
<td>(13231885.3,1.32)</td>
</tr>
<tr>
<td>1992</td>
<td>(258601.4,0.4,0.75)</td>
<td>(24866.7,0.5,0.75)</td>
<td>(13415079.9,0.46)</td>
</tr>
<tr>
<td>1993</td>
<td>(259876.3,1,0.12)</td>
<td>(25445.1,0.9,1)</td>
<td>(13689367.6,0.74)</td>
</tr>
<tr>
<td>1994</td>
<td>(267534.2,0.9,0.5)</td>
<td>(24624.8,0.8,0.75)</td>
<td>(14074004.5,5.5,4.9)</td>
</tr>
<tr>
<td>1995</td>
<td>(283806.6,0.6,0.5)</td>
<td>(24742.4,1.1,1)</td>
<td>(14571273.5,2.5,6)</td>
</tr>
<tr>
<td>1996</td>
<td>(291768.7,0.25,0.45)</td>
<td>(23948.4,1.4,1.75)</td>
<td>(15295511.8,6.5,6)</td>
</tr>
<tr>
<td>1997</td>
<td>(300139.6,0.75,0.25)</td>
<td>(24294.6,1,0.5)</td>
<td>(15790672.9,7.1,6.7)</td>
</tr>
<tr>
<td>1998</td>
<td>(303196.1,9,11)</td>
<td>(22576,0.25,0.5)</td>
<td>(16130665.6,7.5,7)</td>
</tr>
<tr>
<td>2000</td>
<td>(303066.1,1.5)</td>
<td>(24130.9,0.9,0.5)</td>
<td>(16030665.6,7.5,7)</td>
</tr>
<tr>
<td>2001</td>
<td>(33065.1,1.25)</td>
<td>(24780.9,0.9,0.25)</td>
<td>(17020856.4,8,8.1)</td>
</tr>
<tr>
<td>2002</td>
<td>(35555.4,0.75,0.87)</td>
<td>(25691.1,1.25)</td>
<td>(17759834.9,5.8,2.5)</td>
</tr>
<tr>
<td>2003</td>
<td>(379837.1,1)</td>
<td>(26013.8,0.8,0.75)</td>
<td>(18343320.3,8.5,6.8)</td>
</tr>
<tr>
<td>2004</td>
<td>(398234.1,1)</td>
<td>(26653.5,0.25,0.75)</td>
<td>(19063956.2,9.4,9.5)</td>
</tr>
<tr>
<td>2005</td>
<td>(413705.1,1.25)</td>
<td>(26317,1,0.5)</td>
<td>(19743008.8,10,10)</td>
</tr>
<tr>
<td>2006</td>
<td>(445790.6,0.75,0.25)</td>
<td>(26501.3,0.5,0.5)</td>
<td>(20476327.11,13,12)</td>
</tr>
</tbody>
</table>

Table 3: Inputs and output data for labor supply estimate in example 6.2.

<table>
<thead>
<tr>
<th>Year</th>
<th>$X_{i1}$</th>
<th>$X_{i2}$</th>
<th>$Y_{i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1988</td>
<td>(180822.5,11,12)</td>
<td>(19050.6,9,11)</td>
<td>(38.6999,0.11,0.1)</td>
</tr>
<tr>
<td>1989</td>
<td>(189597.1,9,7)</td>
<td>(19264.2,8.8,9.8)</td>
<td>(38.1169,0.13,0.101)</td>
</tr>
<tr>
<td>1990</td>
<td>(218538.7,4.9)</td>
<td>(20611.4,6.6,8.9)</td>
<td>(38.2999,0.134,0.108)</td>
</tr>
<tr>
<td>1991</td>
<td>(245036.4,5.6,7.4)</td>
<td>(22778,4.5,6.9)</td>
<td>(38.0999,0.17,0.12)</td>
</tr>
<tr>
<td>1992</td>
<td>(254059.2,12.2,13.5)</td>
<td>(22859.1,7.9,5)</td>
<td>(37.6999,0.2,0.2)</td>
</tr>
<tr>
<td>1993</td>
<td>(258601.4,12.7,6)</td>
<td>(24866.7,14.6,3)</td>
<td>(37.1999,0.22,0.24)</td>
</tr>
<tr>
<td>1994</td>
<td>(259876.3,11.25,12.75)</td>
<td>(25445,1.6,9.8)</td>
<td>(36.7999,0.28,0.27)</td>
</tr>
<tr>
<td>1995</td>
<td>(267534.2,11,12.2)</td>
<td>(24624.8,12,10)</td>
<td>(36.4,0.31,0.32)</td>
</tr>
<tr>
<td>1996</td>
<td>(283806.6,12.4,11)</td>
<td>(24742.4,1.6,9)</td>
<td>(35.3,0.35,0.37)</td>
</tr>
<tr>
<td>1997</td>
<td>(291768.7,10,9)</td>
<td>(23948.4,4.5,6.8)</td>
<td>(35.0,3.9,0.4)</td>
</tr>
<tr>
<td>1998</td>
<td>(300139.6,6.4,8.2)</td>
<td>(24294.6,9,8.5)</td>
<td>(36.2,0.41,0.43)</td>
</tr>
<tr>
<td>1999</td>
<td>(304941.2,9.8,6.9)</td>
<td>(22576,12,17.5)</td>
<td>(36.6999,0.45,0.48)</td>
</tr>
<tr>
<td>2000</td>
<td>(32069,11,12)</td>
<td>(24638.5,8.9,8.8)</td>
<td>(37.1,0.49,0.51)</td>
</tr>
<tr>
<td>2001</td>
<td>(33065.1,12.15)</td>
<td>(24780,9.9,5.7)</td>
<td>(37.5999,0.54,0.56)</td>
</tr>
<tr>
<td>2002</td>
<td>(35555.4,21.15)</td>
<td>(25691,11,12)</td>
<td>(38.0999,0.59,0.58)</td>
</tr>
<tr>
<td>2003</td>
<td>(379837.16,12)</td>
<td>(26013.8,5.8,25)</td>
<td>(38.4999,0.61,0.62)</td>
</tr>
<tr>
<td>2004</td>
<td>(398234,10,8)</td>
<td>(26653.5,11,12)</td>
<td>(38.9999,0.63,0.66)</td>
</tr>
<tr>
<td>2005</td>
<td>(41705,12.5,11.5)</td>
<td>(26379,9,8.7)</td>
<td>(39.2,0.68,0.68)</td>
</tr>
<tr>
<td>2006</td>
<td>(445790.6,10,25)</td>
<td>(26501,3,0.5,0.5)</td>
<td>(39.3999,0.7,0.71)</td>
</tr>
</tbody>
</table>

\[
\ln \bar{Y}_i = \ln (1.4734,0.0231,0.0342) + (0.1292,0.0087,0.0063)\ln X_{i1} + (-0.1178,0.0075,0.0056)\ln X_{i2} + (0.8861,0.0042,0.0092)\ln \bar{Y}_{i-1}. \tag{38}
\]

\[
\ln \bar{Y}_i = \ln (0.9072,0.0031,0.0085) + (0.0788,0.00065,0.00075)\ln X_{i1} + (-0.1385,0.0021,0.0061)\ln X_{i2} + (0.8613,0.0073,0.0084)\ln \bar{Y}_{i-1}. \tag{39}
\]

Eq.(38) is labor demand and Eq.(39) is labor supply, the coefficient 0.1292 means that, holding $\ln x_2$ and $\ln y_{i-1}$, one percent increase of GDP is predicted to increase $y_i$ by 0.13 percent. The other coefficients have a similar interpretation.

The function form in current research is a simple Cab-Daglas form. We include $\ln x_1$ in equation, because it explains all another variables that are impact on $y_i$. The key point in this article is that newly
estimation by OLS nevertheless all of econometric comments is still, such as stationery time series and Ceteris paribus assumption. More difference of this article is another type of OLS estimation by fuzzy independents and dependent variables. Thus the coefficients are fuzzy and they are not exactly.

7 Summary and Conclusions:
Solving fuzzy linear regression (FLR) and fuzzy nonlinear regression (FNLR) by using universal approximators (UA), that is, FNN is presented in this paper. The problem formulation of the proposed UAM is quite straightforward. To obtain the "Best-approximated" solution of FLRs and FNLRs, the adjustable parameters of FNN are systematically adjusted by using the learning algorithm. In this paper we used Iran labor market data and estimated labor supply and demand. The coefficients are not exact. Because the data of labor market are not exact, then these coefficients are better than traditional OLS estimators.

In this paper, we derived a learning algorithm of fuzzy weights of two-layer feedforward fuzzy neural networks whose input-output relations were defined by extension principle. The effectiveness of the derived learning algorithm was demonstrated by computer simulation of numerical examples. Computer simulation in this paper was performed for two-layer feedforward neural networks using the back-propagation-type learning algorithm. If we use other learning algorithms, we may have different simulation results. For example, some global learning algorithms such as genetic algorithms may train fuzzy connection weights much better than the back-propagation-type learning algorithm.

REFERENCES